

RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	000	FFFFFFFFFFF	FFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	000	FFFFFFFFFFF	FFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	000	FFFFFFFFFFF	FFFFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	000	FFF	FFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	000	FFFFFFFFF	FFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	000	FFFFFFFFF	FFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	000	FFFFFFFFF	FFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		NNN	NNN	000000000	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		NNN	NNN	000000000	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		NNN	NNN	000000000	000	FFF	FFF

```
NN      NN  DDDDDDDD  XX      XX  XX      XX  TTTTTTTTTT  NN      NN
NN      NN  DDDDDDDD  XX      XX  XX      XX  TTTTTTTTTT  NN      NN
NN      NN  DD        DD  XX      XX  XX      XX  TT        TT  NN      NN
NNNN    NN  DD        DD  XX      XX  XX      XX  TT        TT  NNNN    NN
NNNN    NN  DD        DD  XX      XX  XX      XX  TT        TT  NNNN    NN
NN  NN  NN  DD        DD  XX      XX  XX      XX  TT        TT  NN  NN  NN
NN  NN  NN  DD        DD  XX      XX  XX      XX  TT        TT  NN  NN  NN
NN      NN  DD        DD  XX      XX  XX      XX  TT        TT  NN      NN
NN      NN  DD        DD  XX      XX  XX      XX  TT        TT  NN      NN
NN      NN  DD        DD  XX      XX  XX      XX  TT        TT  NN      NN
NN      NN  DDDDDDDD  XX      XX  XX      XX  TT        TT  NN      NN
NN      NN  DDDDDDDD  XX      XX  XX      XX  TT        TT  NN      NN
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

```
0001 0 MODULE NDXXTN (IDENT = 'V04-000'  
0002 0 %BLISS32 [, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE))  
0003 0 ) =  
0004 1 BEGIN  
0005 1  
0006 1 *****  
0007 1 *  
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *  
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *  
0010 1 * ALL RIGHTS RESERVED. *  
0011 1 *  
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *  
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *  
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *  
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *  
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *  
0017 1 * TRANSFERRED. *  
0018 1 *  
0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *  
0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *  
0021 1 * CORPORATION. *  
0022 1 *  
0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *  
0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *  
0025 1 *  
0026 1 *****  
0027 1 *****  
0028 1  
0029 1 ++  
0030 1 FACILITY:  
0031 1 DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility  
0032 1  
0033 1 ABSTRACT: Routines for processing transaction numbers.  
0034 1  
0035 1  
0036 1 ENVIRONMENT: Transportable  
0037 1  
0038 1 AUTHOR: RWF  
0039 1  
0040 1 CREATION DATE: January, 1979  
0041 1  
0042 1 MODIFIED BY:  
0043 1  
0044 1 004 JPK00015 04-Feb-1983  
0045 1 Cleaned up module names, modified revision history to  
0046 1 conform with established standards. Updated copyright dates.  
0047 1  
0048 1 003 JPK00012 24-Jan-1983  
0049 1 Modified NDXVMSMSG.MSG to define error messages for both  
0050 1 DSRINDEX and INDEX.  
0051 1 Added require of NDXVMSREQ.R32 to NDXOUT, NDXFMT, NDXDAT,  
0052 1 INDEX, NDXMSG, NDXXTN, NDXTMS, NDXVMS and NDXPAG for BLISS32.  
0053 1 Since this file defines the error message literals,  
0054 1 the EXTERNAL REFERENCES for the error message literals  
0055 1 have been removed.  
0056 1  
0057 1 002 JPK00008 19-Nov-1982
```



```

: 58      0058 1 |
: 59      0059 1 |
: 60      0060 1 |
: 61      0061 1 |
: 62      0062 1 |
: 63      0063 1 |
: 64      0064 1 |
: 65      0065 1 |
: 66      0066 1 |
: 67      0067 1 |
: 68      0068 1 |
: 69      0069 1 |
: 70      0070 1 |
: 71      0071 1 |
: 72      0072 1 |
: 73      0073 1 |
: 74      0074 1 |
: 75      0075 1 |
: 76      0076 1 |
: 77      0077 1 |
: 78      0078 1 |

```

Changed name of POOL.REQ to DMDEFS.REQ in NDXXTN.

TABLE OF CONTENTS:

FORWARD ROUTINE
ASGXTN : NOVALUE,
XTNPAG;

INCLUDE FILES:

LIBRARY 'NXPORT:XPORT';
SWITCHES LIST (REQUIRE);
REQUIRE 'REQ:PAGEN';

Version: 'V04-000'

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS

ABSTRACT:

A page number carries with it not only its current value, but also
codes as to how those values are to be displayed when they are finally
output. It was decided to do it this way rather than have a separate
table so that the program TCX would have less trouble.

ENVIRONMENT: Transportable BLISS

AUTHOR: Rich Friday

CREATION DATE: 1978

MODIFIED BY:

004 KAD00004 Keith Dawson 07-Mar-1983
Global edit of all modules. Updated module names, idents,
copyright dates. Changed require files to BLISS library.

LITERAL

page_sct_size = 4;

LITERAL

sct_chapt = 1,
sct_index = 2,
sct_append = 3;

Type of section:
! Chapter section.
! Index section.
! Appendix section.

```
: R0136 1
: R0137 1
: R0138 1 LITERAL
: R0139 1     sct_low      = 1;
: R0140 1     sct_high    = 3;
: R0141 1
: R0142 1 MACRO
: R0143 1     sct_typ      = 0, 0, 4, 0 %;
: R0144 1     sct_page_d   = 0, 4, 4, 0 %;
: R0145 1     sct_sub_page = 0, %BPVAL/2, %BPVAL/2, 0 %;
: R0146 1     sct_number   = 1, 0, %BPVAL, 0 %;
: R0147 1     sct_page     = 2, 0, %BPVAL, 0 %;
: R0148 1     sct_subpg_d  = 3, 0, 4, 0 %;
: R0149 1     sct_chapt_d  = 3, 4, 4, 0 %;
: R0150 1     sct_appen_d  = 3, 8, 4, 0 %;
: R0151 1     sct_index_d  = 3, 12, 4, 0 %;
: R0152 1
: R0153 1 MACRO
: R0154 1     sct_run_page = 3, %BPVAL/2, %BPVAL/2, 0 %;
: R0155 1
: R0156 1 MACRO
: R0157 1     page_definition = BLOCK [page_sct_size] %;
: R0158 1
: !
:                               End of PAGEN.REQ
```


NDXXTN
V04-000

0 2
16-Sep-1984 01:16:01
14-Sep-1984 13:07:23

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]NDXXTN.BLI;1

Page 5
(1)

79
80

```
0159 1
0160 1 REQUIRE 'REQ:DMDEFS';
```

NDXX
V04-

[illegible]

Version: 'V04-000'

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

++
FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUSABSTRACT:
Defines literals and macros used in defining, controlling, and
accessing the dynamic memory pool.

ENVIRONMENT: Transportable BLISS

AUTHOR: Rich Friday

CREATION DATE: 1978

MODIFIED BY:

004 KAD00004 Keith Dawson 07-Mar-1983
Global edit of all modules. Updated module names, idents,
copyright dates. Changed require files to BLISS library.

! Structures defining information stored in a dynamic memory pool.

MACRO

POOL = VECTOR [POOL_CNTRL_SIZE] %,
PAD = VECTOR [PAD_CNTRL_SIZE] %;

LITERAL

POOL_CNTRL_SIZE = 3, !Size of POOL control area.
PAD_CNTRL_SIZE = 2; !Size of a Pooled Area Descriptor.

NDXXTN
V04-000

F 2
16-Sep-1984 01:16:01
15-Sep-1984 22:50:50

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[RUNOFF.SRC]DMDEFS.REQ;1 Page 7
(1)

```
: R0218 1 !
: R0219 1 LITERAL Offsets into pool control area (POOL) and pool area descriptor (PAD).
: R0220 1 POOL_MAX_PADS = 0, !Maximum number of PADs that can be accommodated.
: R0221 1 POOL_ACT_PADS = 1, !Current number of allocated PADs.
: R0222 1 POOL_ACT_SIZE = 2; !Number of BPVALS in pool control area.
: R0223 1
: R0224 1 LITERAL
: R0225 1 PAD_SIZE = 0, !Size of pooled area (BLISS VALUES).
: R0226 1 PAD_ADDRESS = 1; !Start of pooled area.
: R0227 1
: R0228 1 ! The GET_SEG_ADDR macro returns the starting address of a segment from the
: R0229 1 ! specified pool.
: R0230 1 MACRO
: MR0231 1 GET_SEG_ADDR(AREA,INDEX) =
: MR0232 1 BEGIN
: MR0233 1 LOCAL
: MR0234 1 PADTAB : REF VECTOR;
: MR0235 1 PADTAB = .AREA+POOL_CNTRL_SIZE*%UPVAL;
: MR0236 1 .PADTAB[PAD_CNTRL_SIZE*(INDEX-1)+PAD_ADDRESS]
: MR0237 1 END
: R0238 1 %;
: R0239 1
: R0240 1 !
! End of DMDEFS.REQ
```

NDXXTN
V04-000

16-Sep-1984 01:16:01
14-Sep-1984 13:07:23

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]NDXXTN.BLI:1

Page 8
(1)

```

: 81      0241 1
: 82      0242 1 REQUIRE 'REQ:XTNTAB';

```

NDXX
V04-

.....

Version: 'V04-000'

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS

ABSTRACT:
Parallel tables for associating index entries and pages.

NOTE: The tables contain one extra entry, which is unused.
That is so subtraction of 1 can be forgotten about.

ENVIRONMENT: Transportable BLISS

AUTHOR: Rich Friday

CREATION DATE: 1978

MODIFIED BY:

002 KAD00002 Keith Dawson 07-Mar-1983
Global edit of all modules. Updated module names, idents,
copyright dates. Changed require files to BLISS library.

LITERAL

max_xtn_count = 100, !Maximum number of transaction numbers (condensed).

!Number of BLISS values in a set of pages.

xtn_pagtab_size = (max_xtn_count + 1) * page_sct_size.

NDXXTN
V04-000

1 2
16-Sep-1984 01:16:01
15-Sep-1984 22:54:49

VAX-11 Bliss-32 V4.0-742
_S255SDUA28:[RUNOFF.SRC]XTNTAB.REQ;1 Page 10
(1)

```
: R0300 1 !Number of BLISS values in a list of transaction numbers.
: R0301 1 !
: R0302 1 !   xtn_xtntab_size = max_xtn_count + 1;
: R0303 1 !
: R0304 1 !MACRO
: R0305 1 !   xtntab_define = VECTOR [xtn_xtntab_size] %,
: R0306 1 !   xpagen_define = BLOCKVECTOR [max_xtn_count + 1, page_sct_size] %;
: R0307 1 !
: R0308 1 !           End of XTNTAB.REQ
```

NDXX
V04-

NDXXTN
V04-000

J 2
16-Sep-1984 01:16:01
14-Sep-1984 13:07:23

VAX-11 BLISS-32 V4.0-742
[RUNOFF.SRC]NDXXTN.BLI;1

Page 11
(1)

```
.. 83
.. 84
.. 85
.. 86
.. 87

L 0309 1
  0310 1 ZIF %BLISS (BLISS32)
  0311 1 %THEN
  0312 1
  0313 1 REQUIRE 'REQ:NDXVMSREQ';
```

NDXX
V04-

: Re

: 2

Version: 'V04-000'

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

FACILITY:
DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility

ABSTRACT:
This file contains external references to the error message numbers
for DSRINDEX/INDEX.

New messages must be defined in NDXVMSMSG.MSG and referenced here:
both in the MACRO section (for DSRINDEX) and the EXTERNAL LITERAL
section (for INDEX)

ENVIRONMENT: VAX/VMS User Mode

AUTHOR: JPK

CREATION DATE: 01-Feb-1983

MODIFIED BY:

004 JPK00022 30-Mar-1983
Modified NDXVMS, NDXFMT, NDXPAG, NDXVMSMSG and NDXVMSREQ
to generate TEX output. Added module NDXTEX.

003 JPK00021 28-Mar-1983
Modified NDXT20 to include E2.0 functionality.
Modified NDXCLIDMP, NDXFMT, NDXPAG, NDXVRS to require RNODEF
for BLISS36 and to remove any conditional require based on
DSRPLUS_DEF.

NDXXTN
V04-000

L 2
16-Sep-1984 01:16:01
15-Sep-1984 22:53:32

VAX-11 BLISS-32 V4.0-742
[RUNOFF.SRC]NDXVMSREQ.R32;1

Page 13
(1)

.. R0371 1
.. R0372 1
.. R0373 1
.. R0374 1
.. R0375 1
.. R0376 1
.. R0377 1

002

JPK00010 04-Feb-1983
Cleaned up module names, modified revision history to
conform with established standards. Updated copyright dates.

REQUIRE 'REQ:RNODEF';

NDXX
V04-

Version: 'V04-000'

* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
* ALL RIGHTS RESERVED. *

* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
* TRANSFERRED. *

* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
* CORPORATION. *

* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *

++
FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUSABSTRACT:
Converts BLISS/VARIANT values into useful names.

ENVIRONMENT: Transportable BLISS

AUTHOR: Rich Friday

CREATION DATE: 1978

MODIFIED BY:

016	KAD00016	Ray Marshall	19-Mar-1984
	Added GERMAN, FRENCH, & ITALIAN.		
015	KAD00015	Keith Dawson	18-Apr-1983
	Made the LN01 conditional the default for vanilla DSR -- its value is 0 (no variant supplied).		
014	KAD00014	Keith Dawson	22-Mar-1983
	Asserted the LN01 conditional when DSRPLUS is asserted.		
013	KAD00013	Keith Dawson	20-Mar-1983
	Removed all references to .BIX and .BTC files.		
012	KAD00012	Keith Dawson	07-Mar-1983
	Global edit of all modules. Updated module names, idents, copyright dates. Changed require files to BLISS library.		

DEFINITION OF /VARIANT BITS

The bit assignments are as follows:

Bit	Weight	Meaning
-----	--------	---------

--	0	If no /VARIANT is supplied (as for vanilla DSR), compile with LN01 support. LN01 support is also implied by the DSRPLUS variant.
----	---	--

0	1	CLEAR = Unassigned SET = Unassigned
---	---	--

1	2	CLEAR = Normal compile SET = Compile for DSRPLUS
---	---	---

4-6	16	CLEAR = English (American) version SET = 16 = German (Austrian) 32 = French 48 = Italian
-----	----	---

This variable (LN01) controls whether or not to compile an LN01-flavored DSR. It is asserted by default, and also whenever DSRPLUS is asserted.

Modules utilizing LN01 are:

DOOPTS NOUT

COMPILETIME

ln01 =
((XVARIANT EQL 0) OR XVARIANT/2)

;

This variable (DSRPLUS) controls compilation for the DSRPLUS program.

All modules utilize DSRPLUS.

COMPILETIME

dsrplus =
(XVARIANT/2)

;

This variable (FLIP) controls compilation of FLIP features of DSRPLUS.
It assures that FLIP features are compiled only on VMS systems.

Modules utilizing FLIP are many and various.

COMPILETIME

flip =

NDXXTN
V04-000

B 3
16-Sep-1984 01:16:01
15-Sep-1984 22:54:08

VAX-11 Bliss-32 V4.0-742
_S255SDUA28:[RUNOFF.SRC]RNODEF.REQ;1

Page 16
(1)

**F1

```

: R0492 2      ( %VARIANT/2 AND %BLISS(BLISS32) )
: R0493 1      ;
: R0494 1
: R0495 1
: R0496 1      -----
: R0497 1      4-6    16    CLEAR =   English (American) version
: R0498 1      SET    =   16 = German (Austrian)
: R0499 1      32 = French
: R0500 1      48 = Italian
: R0501 1      COMPILETIME
: R0502 1      German = ( %VARIANT/16 AND NOT %VARIANT/32 AND NOT %VARIANT/64 ) ;
: R0503 1      COMPILETIME
: R0504 1      French = ( NOT %VARIANT/16 AND %VARIANT/32 AND NOT %VARIANT/64 ) ;
: R0505 1      COMPILETIME
: R0506 1      Italian = ( %VARIANT/16 AND %VARIANT/32 AND NOT %VARIANT/64 ) ;
: R0507 1      -----
:                               End of RNODEF.REQ

```

```

R0508 1
LR0509 1 %IF NOT DSRPLUS
R0510 1 %THEN
R0511 1
R0512 1 MACRO
R0513 1 INDEX$_BADLOGIC = DSRINDEX$_BADLOGIC %
R0514 1 INDEX$_BADVALUE = DSRINDEX$_BADVALUE %
R0515 1 INDEX$_INSVIRMEM = DSRINDEX$_INSVIRMEM %
R0516 1 INDEX$_LINELENG = DSRINDEX$_LINELENG %
R0517 1 INDEX$_NOREF = DSRINDEX$_NOREF %
R0518 1 INDEX$_OPENIN = DSRINDEX$_OPENIN %
R0519 1 INDEX$_OPENOUT = DSRINDEX$_OPENOUT %
R0520 1 INDEX$_TOOMANY = DSRINDEX$_TOOMANY %
R0521 1 INDEX$_VALERR = DSRINDEX$_VALERR %
R0522 1 INDEX$_CANTBAL = DSRINDEX$_CANTBAL %
R0523 1 INDEX$_CLOSEQUOT = DSRINDEX$_CLOSEQUOT %
R0524 1 INDEX$_CONFQUAL = DSRINDEX$_CONFQUAL %
R0525 1 INDEX$_CTRLCHAR = DSRINDEX$_CTRLCHAR %
R0526 1 INDEX$_DOESNTFIT = DSRINDEX$_DOESNTFIT %
R0527 1 INDEX$_DUPBEGIN = DSRINDEX$_DUPBEGIN %
R0528 1 INDEX$_EMPTYIN = DSRINDEX$_EMPTYIN %
R0529 1 INDEX$_IGNORED = DSRINDEX$_IGNORED %
R0530 1 INDEX$_INVINPUT = DSRINDEX$_INVINPUT %
R0531 1 INDEX$_INVRECORD = DSRINDEX$_INVRECORD %
R0532 1 INDEX$_LASTCONT = DSRINDEX$_LASTCONT %
R0533 1 INDEX$_NOBEGIN = DSRINDEX$_NOBEGIN %
R0534 1 INDEX$_NOEND = DSRINDEX$_NOEND %
R0535 1 INDEX$_NOINDEX = DSRINDEX$_NOINDEX %
R0536 1 INDEX$_NOLIST = DSRINDEX$_NOLIST %
R0537 1 INDEX$_OVERSTRK = DSRINDEX$_OVERSTRK %
R0538 1 INDEX$_SKIPPED = DSRINDEX$_SKIPPED %
R0539 1 INDEX$_SYNTAX = DSRINDEX$_SYNTAX %
R0540 1 INDEX$_TEXTFILE = DSRINDEX$_TEXTFILE %
R0541 1 INDEX$_TOODEEP = DSRINDEX$_TOODEEP %
R0542 1 INDEX$_TOOFEW = DSRINDEX$_TOOFEW %
R0543 1 INDEX$_TRUNCATED = DSRINDEX$_TRUNCATED %
R0544 1 INDEX$_COMPLETE = DSRINDEX$_COMPLETE %
R0545 1 INDEX$_CREATED = DSRINDEX$_CREATED %
R0546 1 INDEX$_IDENT = DSRINDEX$_IDENT %
R0547 1 INDEX$_PROCFILE = DSRINDEX$_PROCFILE %
R0548 1 INDEX$_TEXT = DSRINDEX$_TEXT %
R0549 1 INDEX$_TEXTD = DSRINDEX$_TEXTD %
R0550 1 INDEX$_TMS11 = DSRINDEX$_TMS11 %
R0551 1
R0552 1 %FI
R0553 1
R0554 1 EXTERNAL LITERAL
R0555 1 INDEX$_BADLOGIC, <internal logic error detected>
R0556 1 INDEX$_BADVALUE, <'!AS' is an invalid keyword value>
R0557 1 INDEX$_INSVIRMEM, <insufficient virtual memory>
R0558 1 INDEX$_LINELENG, <maximum line length is 120>
R0559 1 INDEX$_NOREF, <page reference not found>
R0560 1 INDEX$_OPENIN, <error opening '!AS' for input>
R0561 1 INDEX$_OPENOUT, <error opening '!AS' for output>
R0562 1 INDEX$_TOOMANY, <too many values supplied>
R0563 1 INDEX$_VALERR, <specified value is out of legal range>
R0564 1 INDEX$_CANTBAL, <can't balance last page>
```

```
.. R0565 1 INDEX$ _CLOSEQUOT, <missing close quote>
.. R0566 1 INDEX$ _CONQUAL, <conflicting qualifiers>
.. R0567 1 INDEX$ _CTRLCHAR, <the following line contains control characters - ignored>
.. R0568 1 INDEX$ _DOESNTFIT, <'!AD' will not fit at the current indentation level>
.. R0569 1 INDEX$ _DUPBEGIN, <duplicate .XPLUS (BEGIN) - inserted as .XPLUS (>>
.. R0570 1 INDEX$ _EMPTYIN, <empty input file '!AS'>
.. R0571 1 INDEX$ _IGNORED, <'!AS' ignored>
.. R0572 1 INDEX$ _INVINPUT, <invalid input file format in file '!AS'>
.. R0573 1 INDEX$ _INVRECORD, <invalid record type in file '!AS'>
.. R0574 1 INDEX$ _LASTCONT, <can't generate continuation heading on last page>
.. R0575 1 INDEX$ _NOBEGIN, <.XPLUS (END) with no .XPLUS (BEGIN) - inserted as .XPLUS (>>
.. R0576 1 INDEX$ _NOEND, <.XPLUS (BEGIN) has no corresponding .XPLUS (END)>
.. R0577 1 INDEX$ _NOINDEX, <no index information in file '!AS'>
.. R0578 1 INDEX$ _NOLIST, <parameter list not allowed>
.. R0579 1 INDEX$ _OVERSTRK, <the following line contains an overstrike sequence>
.. R0580 1 INDEX$ _SKIPPED, <!UL reference!XS inside page range - ignored>
.. R0581 1 INDEX$ _SYNTAX, <error parsing '!AS'>
.. R0582 1 INDEX$ _TEXTFILE, <error processing line !UL of TEX character file '!AS'>
.. R0583 1 INDEX$ _TOODEEP, <maximum subindex depth exceeded>
.. R0584 1 INDEX$ _TOOFEW, <not enough values supplied>
.. R0585 1 INDEX$ _TRUNCATED, <string too long - truncated>
.. R0586 1 INDEX$ _COMPLETE, <processing complete '!AS'>
.. R0587 1 INDEX$ _CREATED, <'!AS' created>
.. R0588 1 INDEX$ _IDENT, <INDEX version !AD>
.. R0589 1 INDEX$ _PROCFILE, <processing file '!AS'>
.. R0590 1 INDEX$ _TEXT, <!AS>
.. R0591 1 INDEX$ _TEXTD, <entry text: '!AD'>
.. R0592 1 INDEX$ _TMS11, <output file full - continuing with file '!AS'>
.. R0593 1
```



```
88 0594 1
89 0595 1 XFI
90 0596 1
91 0597 1 SWITCHES LIST (NOREQUIRE);
92 0598 1
93 0599 1
94 0600 1 MACROS:
95 0601 1
96 0602 1
97 0603 1 EQUATED SYMBOLS:
98 0604 1
99 0605 1
100 0606 1 LITERAL
101 0607 1 TRUE = 1,
102 0608 1 FALSE = 0,
103 0609 1 XTN_MAX_SEGS = 100;
104 0610 1
105 0611 1
106 0612 1
107 0613 1 OWN STORAGE:
108 0614 1
109 0615 1
110 0616 1 EXTERNAL REFERENCES:
111 0617 1
112 0618 1
113 0619 1 EXTERNAL
114 0620 1 XPAGEN : REF XPAGEN_DEFINE,
115 0621 1 XTNCNT,
116 0622 1 XTNLSP : REF PAGE_DEFINITION,
117 0623 1 XTNLSX : REF VECTOR [XTN_MAX_SEGS + 1],
118 0624 1 XTNPOL : REF POOL,
119 0625 1 XTNSGP : REF BLOCK,
120 0626 1 XTNTAB : REF XTNTAB_DEFINE;
121 0627 1
122 0628 1 EXTERNAL ROUTINE
123 0629 1 GPOOL,
124 0630 1 PAGEQL,
125 0631 1 XPOOL;
126 0632 1
```

!Maximum number of pieces into
!which the transaction number
!tables can be broken.

```
128 0633 1 GLOBAL ROUTINE ASGXTN (PAGE, TRANSACTION) : NOVALUE = !
129 0634 1
130 0635 1 ++
131 0636 1 FUNCTIONAL DESCRIPTION:
132 0637 1
133 0638 1 Associates the current page number with a transaction
134 0639 1 number range.
135 0640 1
136 0641 1 FORMAL PARAMETERS:
137 0642 1
138 0643 1 PAGE indicates which page number is to be attached to
139 0644 1 the index entry.
140 0645 1 TRANSACTION, if not zero, is the highest transaction number
141 0646 1 to be associated with the given PAGE.
142 0647 1
143 0648 1 IMPLICIT INPUTS:
144 0649 1
145 0650 1 NONE
146 0651 1
147 0652 1 IMPLICIT OUTPUTS:
148 0653 1
149 0654 1 Implicit in this routine is the compression of the list of
150 0655 1 transaction numbers for a single page. Note that before
151 0656 1 the document page number is copied, a check is made to
152 0657 1 see if the previous transaction number refers to something
153 0658 1 on the same page. If that is the case, then no copy of the
154 0659 1 page number is made.
155 0660 1 This fact is important for the operation of the MODULE
156 0661 1 XPRT, which prints the index entries later.
157 0662 1 If the compression is not made, that module assumes that
158 0663 1 there are distinct pages having the same number; subsequently,
159 0664 1 it won't merge page numbers with 'to' or '-' correctly.
160 0665 1
161 0666 1 ROUTINE VALUE:
162 0667 1 COMPLETION CODES:
163 0668 1
164 0669 1 NONE
165 0670 1
166 0671 1 SIDE EFFECTS:
167 0672 1
168 0673 1 NONE
169 0674 1
170 0675 1 --
171 0676 1
172 0677 2 BEGIN
173 0678 2
174 0679 2 MAP
175 0680 2 PAGE : REF PAGE_DEFINITION;
176 0681 2
177 0682 2 LOCAL
178 0683 2 MERGE;
179 0684 2
180 0685 2 !Is this trip necessary??
181 0686 2
182 0687 2 IF !
183 0688 2 .TRANSACTION EQL 0
184 0689 2 THEN
```

```
RETURN;
!The first time through this code, initialize the pool.
IF .XTNPOL EQL 0
THEN
    !First time through?
    BEGIN
        !First, allocate the pool itself.
        ! (Extra slot gets pointer to XTNSLX segment.)
        GPOOL (XTNPOL, XTN_MAX_SEGS + 1);
        !Now, allocate space for XTNSLX.
        ! (Extra slot avoids having to subtract 1 all the time).
        XTNSLX = XPOOL (XTNPOL, XTN_MAX_SEGS + 1);
    END;

!At this point at least a pool exists for saving the
!segment information. However, the current segment, wherein
!the transaction numbers and associated pages reside,
!may be full, or even not yet allocated.

!In preparation for merging, see if the current page number and
!last referenced page number are the same.
IF
    !
    .XTNLSLSP EQL 0
THEN
    !There is no last page.
    MERGE = FALSE
ELSE
    !Compare the two page numbers, taking display characteristics into account.
    MERGE = PAGEQL (.XTNLSLSP, .PAGE, TRUE);

IF
    !
    .MERGE
THEN
    !The transaction numbers refer to the same page of the
    !document. Just record the new highest transaction number.
    BEGIN
        XTNTAB [.XTNCNT] = .TRANSACTION;      !Record transaction in table, permanently.
        XTNSLX [.XTNPOL [POOL_ACT_PADS]] = .TRANSACTION;    !Remember it for next time around.
    RETURN;
    END;

!The new transaction number does not refer to the last
!page, so no merge was possible. Allocate a new segment
!if the current segment is either full, or else doesn't
!exist.
IF .XTNSGP EQL 0
    OR (.XTNCNT GEQ MAX_XTN_COUNT)
THEN
    !Any segment at all yet?
    !Current segment full?
    BEGIN
        !Allocate a new segment.
        !Note that the transaction numbers and page numbers
        !are saved in the same segment.
        XTNSGP = XPOOL (XTNPOL, XTN_XTNTAB_SIZE + XTN_PAGTAB_SIZE);
        !Make sure that a segment could be allocated.
    IF
        !
```

```
242 0747 4 .XTNSGP EQL 0 OR (.XTNLSX EQL 0) !Catch no XTNLSX space here.
243 0748 4 THEN
244 0749 4 !The requested amount could not be allocated (pool full)
245 0750 4 BEGIN
246 L 0751 4 %IF %BLISS (BLISS32)
247 0752 4 %THEN ! Signal errors for BLISS32
248 0753 4
249 0754 4 SIGNAL_STOP (INDEX$_INSVIRMEM);
250 0755 4
251 U 0756 4 %ELSE ! Use $XPO_PUT_MSG otherwise
252 0757 4
253 0758 4 $XPO PUT MSG (SEVERITY = FATAL,
254 U 0759 4 STRING = 'can't extend transaction pool. ');
255 U 0760 4
256 0761 4 %FI
257 0762 4
258 0763 4 RETURN;
259 0764 4 END;
260 0765 4
261 0766 4 XTNCNT = 0; !No transaction numbers in this segment yet.
262 0767 4 XTNTAB = .XTNSGP; !Transaction table is at start of segment.
263 0768 4 !Page numbers are saved after transaction numbers.
264 0769 4 XPAGEN = .XTNSGP + XTN_XTNTAB_SIZE*XUPVAL;
265 0770 4 END;
266 0771 4
267 0772 4 !At this point, there is definitely a spot free to save the
268 0773 4 transaction number and the associated page number.
269 0774 4 !That slot is the one AFTER the previous slot.
270 0775 4 XTNCNT = .XTNCNT + 1; !New transaction number slot.
271 0776 4 XTNTAB [0] = .XTNCNT; !Remember count in this list.
272 0777 4 XTNLSP = XPAGEN [.XTNCNT, SCT_TYP]; !Remember where this page is.
273 0778 4 BEGIN
274 0779 4 BIND
275 0780 4 COPY = XPAGEN [.XTNCNT, 0,0,0,0] : VECTOR; !Make these structures
276 0781 4 MAP !vectors so that
277 0782 4 PAGE : REF VECTOR; !copying is easier.
278 0783 4 ....
279 0784 4 !Copy items one by one.
280 0785 4 INCR I FROM 0 TO (PAGE_SCT_SIZE -1) DO
281 0786 4 COPY [.I] = .PAGE [.I];
282 0787 4 END;
283 0788 4 !! XPAGEN [.XTNCNT, SCT_TYP] = .PAGE [SCT_TYP]; !Save this page number.
284 0789 4 !! XPAGEN [.XTNCNT, SCT_SUB_PAGE] = .PAGE [SCT SUB PAGE]; !...
285 0790 4 !! XPAGEN [.XTNCNT, SCT_NUMBER] = .PAGE [SCT NUMBER]; !...
286 0791 4 !! XPAGEN [.XTNCNT, SCT_PAGE] = .PAGE [SCT PAGE]; !...
287 0792 4 !! XPAGEN [.XTNCNT, SCT_DISPLAY] = .PAGE [SCT DISPLAY]; !...
288 0793 4 XTNTAB [.XTNCNT] = .TRANSACTION; !Record transaction in table, permanently.
289 0794 4 XTNLSX [.XTNPOL [POOL_ACT_PADS]] = .TRANSACTION; !Remember it for next time around.
290 0795 4 END; !End of ASGXIN
```

```
.TITLE NDXXTN
.IDENT \V04-000\

.EXTRN DSRINDEX$_BADLOGIC
.EXTRN DSRINDEX$_BADVALUE
.EXTRN DSRINDEX$_INSVIRMEM
```



```
.EXTRN DSRINDEX$_LINELENG
.EXTRN DSRINDEX$_NOREF
.EXTRN DSRINDEX$_OPENIN
.EXTRN DSRINDEX$_OPENOUT
.EXTRN DSRINDEX$_TOOMANY
.EXTRN DSRINDEX$_VALERR
.EXTRN DSRINDEX$_CANTBAL
.EXTRN DSRINDEX$_CLOSEQUOT
.EXTRN DSRINDEX$_CONFQUAL
.EXTRN DSRINDEX$_CTRLCHAR
.EXTRN DSRINDEX$_DOESNTFIT
.EXTRN DSRINDEX$_DUPBEGIN
.EXTRN DSRINDEX$_EMPTYIN
.EXTRN DSRINDEX$_IGNORED
.EXTRN DSRINDEX$_INVINPUT
.EXTRN DSRINDEX$_INVRECORD
.EXTRN DSRINDEX$_LASTCONT
.EXTRN DSRINDEX$_NOBEGIN
.EXTRN DSRINDEX$_NOEND
.EXTRN DSRINDEX$_NOINDEX
.EXTRN DSRINDEX$_NOLIST
.EXTRN DSRINDEX$_OVERSTRK
.EXTRN DSRINDEX$_SKIPPED
.EXTRN DSRINDEX$_SYNTAX
.EXTRN DSRINDEX$_TEXTFILE
.EXTRN DSRINDEX$_TOODEEP
.EXTRN DSRINDEX$_TOOFEW
.EXTRN DSRINDEX$_TRUNCATED
.EXTRN DSRINDEX$_COMPLETE
.EXTRN DSRINDEX$_CREATED
.EXTRN DSRINDEX$_IDENT
.EXTRN DSRINDEX$_PROCFILE
.EXTRN DSRINDEX$_TEXT, DSRINDEX$_TEXTD
.EXTRN DSRINDEX$_TMS11
.EXTRN XPAGEN, XTNCNT, XTNLSP
.EXTRN XTNLSP, XTNPOL, XTNSGP
.EXTRN XTNTAB, GPOOL, PAGEQL
.EXTRN XPOOL
```

```
.PSECT $CODE$,NOWRT,2
```

```
OFFC 00000
5B 00000000G EF 9E 00002
5A 00000000G EF 9E 00009
59 00000000G EF 9E 00010
58 00000000G EF 9E 00017
57 00000000G EF 9E 0001E
56 00000000G EF 9E 00025
55 00000000G EF 9E 0002C
54 00000000G EF 9E 00033
53      08 AC D0 0003A
      01 12 0003E
      65 D5 00041 18:
      19 12 00043
7E      65 8F 9A 00045
```

```
.ENTRY ASGXTN, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,- R11 : 0633
MOVAB XPAGEN, R11
MOVAB XTNLSP, R10
MOVAB XPOOL, R9
MOVAB XTNLSP, R8
MOVAB XTNSGP, R7
MOVAB XTNTAB, R6
MOVAB XTNPOL, R5
MOVAB XTNCNT, R4
MOVL TRANSACTION, R3 : 0688
BNEQ 1$
RET
TSTL XTNPOL : 0694
BNEQ 2$
MOVZBL #101, -(SP) : 0699
```

00000000G	EF		55	DD	00049	PUSHL	R5		
	7E	65	02	FB	0004B	CALLS	#2, GP00L		
			8F	9A	00052	MOVZBL	#101, -(SP)		0702
	69		55	DD	00056	PUSHL	R5		
	68		02	FB	00058	CALLS	#2, XPOOL		
	51		50	D0	0005B	MOVL	R0, XTNLSP		
			6A	D0	0005E	MOVL	XTNLSP, R1		0713
			04	12	00061	BNEQ	3\$		
			50	D4	00063	CLRL	MERGE		0716
			0E	11	00065	BRB	4\$		
			01	DD	00067	PUSHL	#1		0719
		04	AC	DD	00069	PUSHL	PAGE		
			51	DD	0006C	PUSHL	R1		
00000000G	EF		03	FB	0006E	CALLS	#3, PAGEQL		
	0A		50	E9	00075	BLBC	MERGE, 5\$		0722
	50		64	D0	00078	MOVL	XTNCNT, R0		0727
00 B640			53	D0	0007B	MOVL	R3, @XTNTAB[R0]		
			5F	11	00080	BRB	11\$		0728
			67	D5	00082	TSTL	XTNSGP		0736
			09	13	00084	BEQL	6\$		
00000064	8F		64	D1	00086	CMPL	XTNCNT, #100		0737
			2E	19	0008D	BLSS	9\$		
	7E	01F9	8F	3C	0008F	MOVZWL	#505, -(SP)		0743
			55	DD	00094	PUSHL	R5		
	69		02	FB	00096	CALLS	#2, XPOOL		
	67		50	D0	00099	MOVL	R0, XTNSGP		
			04	13	0009C	BEQL	7\$		0747
			68	D5	0009E	TSTL	XTNLSP		
			0E	12	000A0	BNEQ	8\$		
		00000000G	8F	DD	000A2	PUSHL	#DSRINDEX\$, INSVIRMEM		0754
00000000G	00		01	FB	000A8	CALLS	#1, LIB\$STOP		
				04	000AF	RET			0750
	66		64	D4	000B0	CLRL	XTNCNT		0766
6B	67	00000194	67	D0	000B2	MOVL	XTNSGP, XTNTAB		0767
			8F	C1	000B5	ADDL3	#404, XTNSGP, XPAGEN		0769
			64	D6	000BD	INCL	XTNCNT		0775
	51		64	D0	000BF	MOVL	XTNCNT, R1		0776
	00 B6		51	D0	000C2	MOVL	R1, @XTNTAB		
52	51		04	78	000C6	ASHL	#4, R1, R2		0777
	52		68	C0	000CA	ADDL2	XPAGEN, R2		
	6A		52	D0	000CD	MOVL	R2, XTNLSP		
			50	D4	000D0	CLRL	1		0786
	6240	04 BC40	00	D0	000D2	MOVL	@PAGE[1], (R2)[1]		
F6	50		03	F3	000D8	AOBLEQ	#3, 1, 10\$		
	00 B641		53	D0	000DC	MOVL	R3, @XTNTAB[R1]		0793
	50		65	D0	000E1	MOVL	XTNPOL, R0		0794
	50		04	C0	000E4	ADDL2	#4, R0		
	50		60	D0	000E7	MOVL	(R0), R0		
	00 B840		53	D0	000EA	MOVL	R3, @XTNLSP[R0]		
			04	000EF	RET				0795

; Routine Size: 240 bytes, Routine Base: \$CODE\$ + 0000

; 291 0796 1

```

293 0797 1 GLOBAL ROUTINE XTNPAG (TRANSACTION) = !
294 0798 1
295 0799 1 ++
296 0800 1 FUNCTIONAL DESCRIPTION:
297 0801 1
298 0802 1 Given a transaction number, return the address of
299 0803 1 the corresponding page number.
300 0804 1
301 0805 1 FORMAL PARAMETERS:
302 0806 1
303 0807 1 TRANSACTION - The transaction number.
304 0808 1
305 0809 1 IMPLICIT INPUTS:
306 0810 1
307 0811 1 NONE
308 0812 1
309 0813 1 IMPLICIT OUTPUTS:
310 0814 1
311 0815 1 NONE
312 0816 1
313 0817 1 ROUTINE VALUE:
314 0818 1 COMPLETION CODES:
315 0819 1
316 0820 1 Address of the corresponding page number.
317 0821 1
318 0822 1 SIDE EFFECTS:
319 0823 1
320 0824 1 NONE
321 0825 1
322 0826 1 --
323 0827 1
324 0828 2 BEGIN
325 0829 2
326 0830 2 IF !
327 0831 2 .TRANSACTION EQL 0
328 0832 2 THEN
329 0833 2 RETURN 0;
330 0834 2
331 0835 2 !find the correct segment number.
332 0836 2 !NOTE: Start at 2 because first is XTNSLX.
333 0837 2
334 0838 2 INCR I FROM 2 TO .XTNPOL [POOL_ACT_PADS] DO
335 0839 2 BEGIN
336 0840 2
337 0841 2 IF !
338 0842 2 .TRANSACTION LEQ .XTNSLX [.I]
339 0843 2 THEN
340 0844 2 !Search segment for exact transaction number.
341 0845 2 !That results in an index into the corresponding
342 0846 2 !set of saved pages.
343 0847 2 BEGIN
344 0848 2
345 0849 2 LOCAL
346 0850 2 XTN TABLE : REF XTNTAB DEFINE,
347 0851 2 XPAGEN : REF XPAGEN_DEFINE;
348 0852 2
349 0853 2 XTN_TABLE = GET_SEG_ADDR (XTNPOL, .I);

```

```

350      0854      4      XPAGEN = GET_SEG_ADDR (XTNPOL, .I) + XTN_XTNTAB_SIZE*%UPVAL;
351      0855      4
352      0856      4      INCR J FROM 1 TO .XTN_TABLE [0] DO
353      0857      3      BEGIN
354      0858      3
355      0859      3      IF
356      0860      3      .TRANSACTION LEQ .XTN_TABLE [.J]
357      0861      3      THEN
358      0862      3      RETURN XPAGEN [.J, SCT_TYP]
359      0863      3
360      0864      4      END;
361      0865      4
362      0866      4      END;
363      0867      4
364      0868      2      END;
365      0869      2
366      L 0870      2      %IF %BLISS (BLISS32)
367      0871      2      %THEN
368      0872      2      ! Signal errors for BLISS32
369      0873      2      SIGNAL_STOP (INDEX$_NOREF, 0, INDEX$_BADLOGIC);
370      0874      2
371      U 0875      2      %ELSE
372      0876      2      ! Use $XPO_PUT_MSG otherwise
373      0877      2      $XPO_PUT_MSG (SEVERITY = FATAL,
374      0878      2      STRING = 'internal error - page reference not found.');
```

				.EXTRN	PAGEN	
			007C 00000	.ENTRY	XTNPAG, Save R2,R3,R4,R5,R6	: 0797
	56	04	AC D0 00002	MOVL	TRANSACTION, R6	: 0831
			68 13 00006	BEQL	5\$	
	54	00000000G	EF D0 00008	MOVL	XTNPOL, R4	: 0838
	51		01 D0 0000F	MOVL	#1, I	: 0842
			3A 11 00012	BRB	4\$	
	00000000GFF41		56 D1 00014 1\$:	CMPL	P6, @XTNLSX[I]	
			30 14 0001C	BGTR	4\$	
	53	0C	A4 9E 0001E	MOVAB	12(R4), R3	: 0853
	52		53 D0 00022	MOVL	R3, PADTAB	
50	51		01 78 00025	ASHL	#1, I, R0	
	55	FC	A240 D0 00029	MOVL	-4(PADTAB)[R0], XTN_TABLE	
52	FC A340	00000194	8F C1 0002E	ADDL3	#404, -4(PADTAB)[R0], XPAGEN	: 0854
			53 D4 00038	CLRL	J	: 0856
			0E 11 0003A	BRB	3\$	
	6543		56 D1 0003C 2\$:	CMPL	R6, (XTN_TABLE)[J]	: 0860
			08 14 00040	BGTR	3\$	

50	53	04	78	00042	ASHL	#4, J, R0	0862
	50	52	C0	00046	ADDL2	XPAGEN, R0	
			04	00049	RET		
EE	53	65	F3	0004A	3\$: AOBLEQ	(XTN TABLE), J, 2\$	0857
C1	51	A4	F3	0004E	4\$: AOBLEQ	4(R4), I, 1\$	0838
		8F	DC	00053	PUSHL	#DSRINDEX\$_BADLOGIC	0873
	00000000G	7E	D4	00059	CLRL	-(SP)	
		8F	DD	0005B	PUSHL	#DSRINDEX\$_NOREF	
00000000G	00	03	FB	00061	CALLS	#3, LIB\$STOP	
	50	00000000G	EF	9E	MOVAB	PAGEN, R0	0882
			04	0006F	RET		
		50	D4	00070	5\$: CLRL	R0	0888
			04	00072	RET		

; Routine Size: 115 bytes, Routine Base: \$CODE\$ + 00F0

385
386
387
388

0889 1
0890 1 END
0891 1
0892 0 ELUDOM

!End of module

.EXTRN LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	355	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	0	0	252	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:NDXXTN/OBJ=OBJ\$:NDXXTN MSRC\$:NDXXTN/UPDATE=(ENH\$:NDXXTN)

Size: 355 code + 0 data bytes
Run Time: 00:11.1
Elapsed Time: 00:25.2
Lines/CPU Min: 4830

NDXXTN
V04-000

N 3
16-Sep-1984 01:16:01

VAX-11 Bliss-32 V4.0-742

Page 28

: Lexemes/CPU-Min: 12043
: Memory Used: 88 pages
: Compilation Complete

NM
V04-

0346 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

NEWPG LIS	NODOPX LIS	OFT LIS	OUTXT LIS
NDXURS LIS	NOTE LIS	OUTLN LIS	PACK LIS
NM LIS	OUTXHR LIS	OUTCHA LIS	OUTHOR LIS
NOXXTN LIS			